Building Simulation 2003

Eighth International IBPSA Conference
Eindhoven, Netherlands
August 11-14, 2003

A CONVERGENT OPTIMIZATION METHOD USING PATTERN SEARCH ALGORITHMS WITH ADAPTIVE PRECISION SIMULATION

Michael Wetter[1] and Elijah Polak[2].
[1]Simulation Research Group, Building Technologies Department,
Environmental Energy Technologies Division, Lawrence Berkeley National Laboratory,
Berkeley, CA 94720, USA.
[2]Department of Electrical Engineering, University of California at Berkeley,
Berkeley, CA 94720, USA.

ABSTRACT

Thermal building simulation programs, such as EnergyPlus, approximate solutions of a differential algebraic system of equations. While the theoretical solution is usually continuously differentiable in the building design parameters, the approximate solutions may not even be continuous, due to adaptive variations in solver iterations and the use of adaptive integration meshes. Hence, when a smooth cost function, defined on the design parameters, is evaluated using a thermal building simulation program, it becomes replaced with an approximation that fails to be even continuous. Consequently, when used in conjunction with an optimization algorithm that depends on smoothness of the cost function, the algorithm is quite likely to jam at a non-optimal point. Obviously, in such situations, the potential economic gains that optimization offers are not attained.

As an illustration, we present an example, using the EnergyPlus whole building energy simulation program to evaluate our cost function, in which the Hooke-Jeeves algorithm terminates at a non-stationary point. To prevent such failures, we have developed an adaptive simulation precision control algorithm that can be used in conjunction with a family of derivative free optimization algorithms. The resulting composite algorithms are demonstrably convergent to an exact stationary point. We present the main ingredients of the composite algorithms and show by numerical experiments that using coarse approximations in the early iterations can significantly reduce the computation time.

INTRODUCTION

We propose a new approach for the optimization of a cost function $f\colon \mathbb{R}^n \to \mathbb{R}$ that must be defined on the solutions of a coupled system of differential algebraic equations (DAE). We assume that the coupled system of DAE has a unique solution which is continuously differentiable in the design parameter, but that the solution can only be approximated numerically using computationally expensive simulations. We assume that the termination criteria of the numerical solvers depend on the design parameters. In this situation, a computer code for solving these systems usually defines a numerical approximation function $f^*(\varepsilon,\cdot)$, where $\varepsilon \in \mathbb{R}^q_+$ denotes the precision parameter of the DAE solvers, which is discontinuous in the design parameter. This situation is typical of many problems in system design, optimal control, system identification, etc.

In the past, both deterministic nonlinear optimization algorithms and probabilistic optimization algorithms have been used heuristically in conjunction with such approximating functions problems (such as in Wetter 2001, Wright and Loosemore 2001, Caldas and Norford 2002). In such situations, deterministic nonlinear optimization algorithms that use fixed precision function evaluations may yield only partial improvement due to failure far from an optimum point, and probability based optimization methods may require a prohibitively large number of function evaluations to achieve convergence with high probability. Both of these optimization approaches are usually employed in conjunction with high precision function evaluations for all iterations, which results in long computation time, and little can be said about their convergence properties.

In this paper, we present a new optimization algorithm that combines a Generalized Pattern Search (GPS) optimization algorithm (Audet and Dennis 2003) with an adaptive test that determines the precision with which the cost functions $\{f^*(\varepsilon,\cdot)\}_{\varepsilon\in\mathbb{R}^q_+}$ must be evaluated. As a result, our algorithm uses coarse approximations in the early iterations, with the precision gradually increased as a solution is approached. Such adaptive schemes are known to yield significant reductions in computation time (Polak 1997) over fixed precision schemes.

For continuously differentiable cost functions $f(\cdot)$, our GPS algorithms with adaptive precision function evaluations construct a sequence of iterates that converge to stationary accumulation points. We begin by showing that the cost function $f(\cdot)$ is continuously differentiable for many problems in building and HVAC optimization. Then, we explain the main ingredients of GPS algorithms. Next we explain why optimization using fixed precision cost function evaluations can fail and give an example of failure. Finally, we state the assumptions on the simulation model for our method to converge and give an example of how the cost function evaluation error is controlled. We close with a numerical example that uses adaptive precision cost function evaluations.

MINIMIZATION PROBLEM

We will consider problems of the form

$$\min_{x \in \mathbb{R}^n} f(x), \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a once continuously differentiable cost function. For some $m, l \in \mathbb{N}$, we assume that $f(\cdot)$ is of the form

$$f(x) \triangleq F\big(z(x,1)\big), \tag{2}$$

where $F\colon \mathbb{R}^m \to \mathbb{R}$ is a once continuously differentiable function that is defined on the solutions of the coupled system of differential algebraic (DAE) equations

$$\frac{dz(x,t)}{dt} = h\big(x,\mu\big), \qquad t \in [0,\ 1], \tag{3a}$$

$$z(x,0) = 0, \tag{3b}$$

$$\gamma\big(x, z(x,t), \mu\big) = 0, \qquad t \in [0,\ 1], \tag{3c}$$

where $h\colon \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^m$ and $\gamma\colon \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^l$ are once continuously differentiable in all arguments. We assume that (3c) has for all arguments a unique solution $\mu \in \mathbb{R}^l$ and that the matrix $d\gamma(\cdot,\cdot,\mu)/d\mu \in \mathbb{R}^{l\times l}$ is non-singular. By using the Implicit Function Theorem (Polak 1997) and standard theory of differential equations (Coddington and Levinson 1955), it can be shown that for all $x \in \mathbb{R}^n$ and all $t \in [0,\ 1]$, (3) has a unique, once continuously differentiable solution $z(x,t)$. Thus, $f(x)$ is once continuously differentiable.

(3) is a typical system of equations that is solved during a thermal building simulation after the spatial domain of wall, floor and ceiling constructions has been discretized in a finite number of nodal points. For example, the components of the vector $z(\cdot,\cdot)$ can be the zonal air temperature, the solid temperature at the nodal points, and the building energy consumption, and $\gamma(\cdot,\cdot,\cdot)$ can be a system of nonlinear equations, such as used in describing convective heat transfer.

For simplicity, we assume (1) is unconstrained. The case with linear constraints on x is discussed in Audet and Dennis (2003) and Polak and Wetter (2003), and constraints on independent parameters can be added by using barrier or penalty methods as described in Bertsekas (1999).

We assume that $z(x,t)$ cannot be evaluated exactly, but that it can be approximated by functions $z^*(\varepsilon,x,t)$, $z^*\colon \mathbb{R}^q_+ \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$, where $\varepsilon \in \mathbb{R}^q_+$ is a vector that contains the precision parameters of the DAE solvers. For example, given a design parameter $x \in \mathbb{R}^n$, $z^*(\varepsilon,x,t)$ is the approximate solution to $z(x,t)$ of (3) as evaluated by a computer simulation program with solver precision parameters ε. Thus, we must define an approximating cost function $f^*(\varepsilon,x) \triangleq F(z^*(\varepsilon,x,1))$ which is in general discontinuous in x.

OPTIMIZATION ALGORITHM

Since $f^*(\varepsilon,\cdot)$ is discontinuous, $f^*(\varepsilon,\cdot)$ cannot be differentiated. Therefore, we will use GPS algorithms, which are derivative-free optimization algorithms. For $k \in \mathbb{N}$, let $x_k \in \mathbb{R}^n$ denote the current iterate. GPS algorithms have in common that they search for a lower function value than $f^*(\varepsilon,x_k)$ on the points $x_k \pm \Delta_k e_i$, $i \in \{1,\ldots,n\}$, where Δ_k is a positive mesh size factor. Each GPS algorithm has a rule that selects a finite number of points on a mesh defined by $\mathbb{M}(x_0,\Delta_k) \triangleq \{x_0 + \Delta_k e_i m \mid i \in \{1,\ldots,n\}, m \in \mathbb{Z}\}$, where $x_0 \in \mathbb{R}^n$ denotes the initial iterate. It is this rule that distinguishes the different GPS algorithms, such as the Hooke-Jeeves algorithm or the Coordinate Search.

If a point x' on the mesh $\mathbb{M}(x_0,\Delta_k)$ with lower cost than $f^*(\varepsilon,x_k)$ has been found, then the search continues at x' with the same mesh size factor Δ_k. Otherwise, the search continues at x_k with a reduced mesh size factor Δ_k. If Δ_k is smaller than a user-specified limit, the search stops.

Torczon (1997) and Audet and Dennis (2003) proved that for once continuously differentiable cost functions that are radially unbounded, any GPS algorithm converges to a point x^* that satisfies the first order optimality condition $\nabla f(x^*) = 0$.

FIXED PRECISION FUNCTION EVALUATION

We now discuss the situation where the precision of the approximating cost function $f^*(\varepsilon,\cdot)$ cannot be controlled, i.e., ε is fixed for all iterations.

In thermal building simulation programs, such as in EnergyPlus (Crawley et al. 2001), TRNSYS (Klein, Duffie, and Beckman 1976), DOE-2 (Winkelmann et al. 1993), etc., the solvers for the partial differential equations, the ordinary differential equations, and the algebraic equations are implemented in a way that makes it impossible to establish error bounds for the approximating solution of the differential equations, and hence, for the approximating cost function. For example, to simulate a thermal zone with daylighting control and purchased heating and cooling, EnergyPlus uses at least ten precision parameters, most of which are fixed at compile time.

For coupled systems of equations, adjusting the precision parameters independently of each other can cause divergence. Because of the way in which building simulation programs are written, it seems to be impossible to establish a scheme for adjusting all precision parameters and obtaining an error bound for the approximating solutions $\{z^*(\varepsilon,\cdot,\cdot)\}_{\varepsilon\in\mathbb{R}^q_+}$ and hence for the approximating cost functions $\{f^*(\varepsilon,\cdot)\}_{\varepsilon\in\mathbb{R}^q_+}$. Thus, precision control is not applicable, and ε must be fixed for such systems.

It is common in building simulation programs that the termination criteria of the solvers depend on x. Then, a perturbation of x can cause a change in the sequence of solver iterations, which causes $f^*(\varepsilon,x)$ to be discontinuous in x. Furthermore, if variable step size integration methods are used, then the integration mesh can change from one simulation to the next. Therefore, part of the change in function values between different points is caused by a combination of a change in the number of solver iterations and a change in the integration mesh. Consequently, $f^*(\varepsilon,\cdot)$ is discontinuous, and a descent direction of $f^*(\varepsilon,\cdot)$ may not be a descent direction for $f(\cdot)$. Consequently, optimization using fixed precision function evaluations can fail.

As an example of this, Figure 1 shows the normalized source energy consumption for cooling and lighting energy from July 1 to July 7 for the office building presented in Wetter (2001). The computations were done with EnergyPlus 1.0.2 for Linux. The independent parameters are the widths of the west and east windows. The dots show the iteration steps of the Hooke-Jeeves method, which fails at the non-optimal point in the left of the figure due to a discontinuity of $f^*(\varepsilon,\cdot)$.

Since it does not seem to be possible to control precision with today's building simulation programs, one can only hope to come close to an optimal point of $f(\cdot)$. The risk of failing at a discontinuity far from an optimal point can be reduced – but not eliminated – by selecting a large initial step size Δ_0 and using tight precision ε. However, there is no recipe to say how large Δ_0 should be in general and selecting ε too small may result in prohibitively long computation time.

To guarantee convergence to a point x^* that satisfies $\nabla f(x^*) = 0$, one needs to use adaptive precision function evaluation as described in the next section.

ADAPTIVE PRECISION FUNCTION EVALUATION

We will assume that $f(\cdot)$ and its approximating functions $\{f^*(\varepsilon,\cdot)\}_{\varepsilon\in\mathbb{R}^q_+}$ have following properties:

Assumption 1

1. *We know an error bound function $\varphi\colon \mathbb{R}^q_+ \to \mathbb{R}_+$ such that for any bounded set $\mathbf{S} \subset \mathbb{R}^n$, there exist an $\varepsilon_{\mathbf{S}} \in \mathbb{R}^q_+$ and a scalar $K_{\mathbf{S}} \in (0,\infty)$ such that for all $x \in \mathbf{S}$ and for all $\varepsilon \in \mathbb{R}^q_+$, with $\varepsilon < \varepsilon_{\mathbf{S}}$* [1],

$$|f^*(\varepsilon,x) - f(x)| \le K_{\mathbf{S}}\,\varphi(\varepsilon). \tag{4}$$

Furthermore,

$$\lim_{\|\varepsilon\|\to 0} \varphi(\varepsilon) = 0. \tag{5}$$

2. *The function $f\colon \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable.* □

The functions $\{f^*(\varepsilon,\cdot)\}_{\varepsilon\in\mathbb{R}^q_+}$ may be discontinuous. Notice that we need to know a bound of the function $\varphi(\cdot)$, but not the constant $K_{\mathbf{S}}$.

Assumption 1 ensures that we can reduce the approximation error to an arbitrarily small value. See Brenan, Campbell, and Petzold (1989), Hairer and Wanner (1996) or Ascher and Petzold (1998) for numerical methods that generate approximate solutions to (3) for which error bounds can be obtained.

One of the authors is currently developing a thermal building simulation program that allows satisfying Assumption 1.

Next, we state an assumption on the level sets of the family of approximating functions. To do so, we first define the notion of a level set.

Definition 1 (Level Set) *Given a function $f\colon \mathbb{R}^n \to \mathbb{R}$ and an $\alpha \in \mathbb{R}$, such that $\alpha \ge \inf_{x\in\mathbb{R}^n} f(x)$, we will say that the set $\mathbf{L}_\alpha(f) \subset \mathbb{R}^n$, defined as*

$$\mathbf{L}_\alpha(f) \triangleq \{x \in \mathbb{R}^n \mid f(x) \le \alpha\}, \tag{6}$$

is a level set of $f(\cdot)$, parametrized by α. □

Assumption 2 (Compactness of Level Sets) *Let $\{f^*(\varepsilon,\cdot)\}_{\varepsilon\in\mathbb{R}^q_+}$ be as in Assumption 1, let $x_0 \in \mathbb{R}^n$ be the initial iterate, and let $\varepsilon_0 \in \mathbb{R}^q_+$ be the initial solver precision parameter. We assume that there exists a compact set $\mathbf{C} \subset \mathbb{R}^n$ such that for all $\varepsilon \in \mathbb{R}^q_+$, with $\varepsilon < \varepsilon_0$,*

$$\mathbf{L}_{f^*(\varepsilon_0,x_0)}\big(f^*(\varepsilon,\cdot)\big) \subset \mathbf{C}. \tag{7}$$

□

[1] For $\varepsilon \in \mathbb{R}^q$, by $\varepsilon < \varepsilon_{\mathbf{S}}$, we mean that $\varepsilon^i < \varepsilon^i_{\mathbf{S}}$, for all $i \in \{1,\ldots,q\}$.

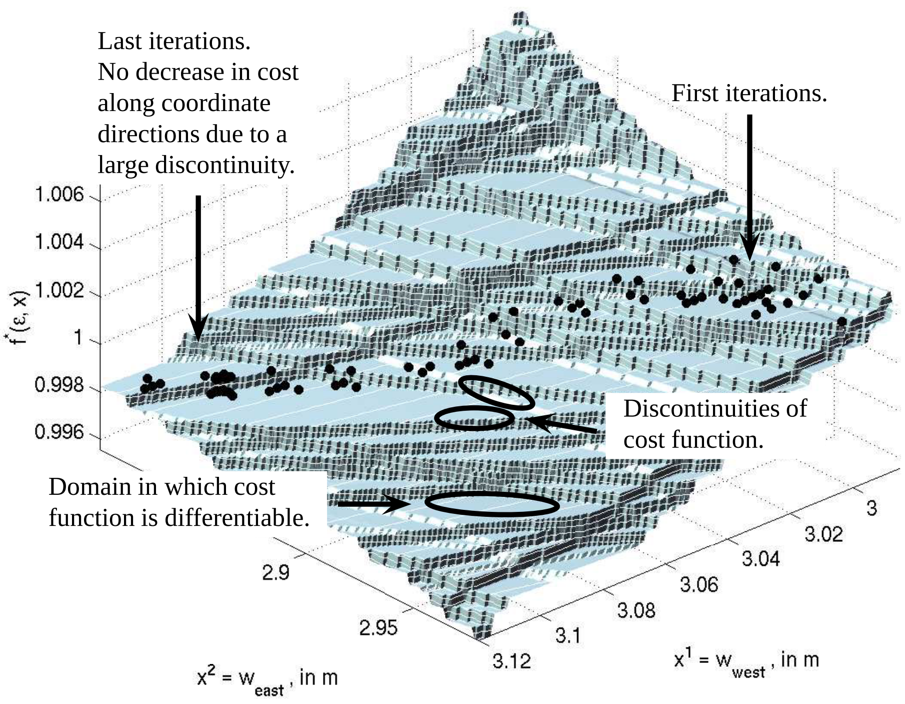

Figure 1: Parametric plot of the normalized source energy consumption for cooling and lighting as a function of the width of the west and east facing window. The dots are iterates of the Hooke-Jeeves algorithm.

Assumption 2 ensures that for a fixed mesh size factor Δ_k, there are only a finite number of mesh points that are candidates for a decrease in $f^*(\varepsilon,\cdot)$. Therefore, the GPS algorithms will reduce the mesh size factor Δ_k after a finite number of iterations.

To control precision of the approximating cost functions $f^*(\cdot,\cdot)$, we need to define a function $\rho\colon \mathbb{R}_+ \to \mathbb{R}^q_+$ that assigns the solver precision parameter ε as a function of the mesh size factor Δ_k for $k \in \mathbb{N}$. The function $\rho(\cdot)$ must be such that the composition $\varphi(\rho(\cdot))$ is strictly monotone decreasing and satisfies $\varphi(\rho(\Delta))/\Delta \to 0$, as $\Delta \to 0$.

For example, suppose that $\varphi(\varepsilon) = \|\varepsilon\|$. Then, we can define $\rho(\Delta) \triangleq \Delta^\alpha \widehat{\varepsilon}$, where $\widehat{\varepsilon} \in \mathbb{R}^q_+$ is fixed and $\alpha > 1$ is used to control how fast the precision is increased. Then,

$$\lim_{\Delta\to 0} \frac{\varphi(\rho(\Delta))}{\Delta} = \lim_{\Delta\to 0} \frac{\varphi(\Delta^\alpha \widehat{\varepsilon})}{\Delta} = \lim_{\Delta\to 0} \frac{\Delta^\alpha}{\Delta}\|\widehat{\varepsilon}\| = 0. \quad (8)$$

In Algorithm 1, we show the implementation of the Coordinate Search algorithm with adaptive precision function evaluations. We selected the Coordinate Search algorithm because it is the simplest member of the family of GPS algorithms and illustrates best how the precision control must be implemented. In Polak and Wetter (2003), we show a generic model GPS algorithm and the implementation of the Hooke-Jeeves algorithm.

Algorithm 1

```
Data:    Initial iterate $x_0 \in \mathbf{X}$;
         Initial mesh size factor $\Delta_0 \in \mathbb{Q}$, $\Delta_0 > 0$.
Maps:    Function $\rho\colon \mathbb{R}_+ \to \mathbb{R}^q_+$ (to assign $\varepsilon$),
         such that the composition
         $\varphi \circ \rho\colon \mathbb{R}_+ \to \mathbb{R}_+$
         is strictly monotone decreasing and
         $\varphi(\rho(\Delta))/\Delta \to 0$, as $\Delta \to 0$.
Step 0:  Initialize $k = 0$, and $\varepsilon = \rho(\Delta_0)$.
Step 1:  Search
         For $i \in \{1,\dots,n\}$,
             Set $x' = x_k + \Delta_k e_i$.
             If $f^*(\varepsilon,x') < f^*(\varepsilon,x_k)$, go to Step 3.
             Set $x' = x_k - \Delta_k e_i$.
             If $f^*(\varepsilon,x') < f^*(\varepsilon,x_k)$, go to Step 3.
         end for.
Step 2:  No improvement found.
         Set $x_{k+1} = x_k$, $\Delta_{k+1} = \Delta_k/2$,
         $\varepsilon = \rho(\Delta_{k+1})$, and go to Step 4.
Step 3:  Reduced cost.
         Set $x_{k+1} = x'$, $\Delta_{k+1} = \Delta_k$, and
         go to Step 4.
Step 4:  Replace $k$ by $k+1$, and go to Step 1.
```

Polak and Wetter (2003) showed that under Assumption 1 and Assumption 2 any sequence of iterates generated by Algorithm 1 contains an accumulation point $x^* \in \mathbb{R}^n$ that satisfies the first order optimality condition $\nabla f(x^*) = 0$. Thus, the precision control guarantees that GPS algorithms do not converge to discontinuities of $f^*(\varepsilon,\cdot)$. An important feature of the adaptive precision scheme is the use of coarse approximations in the early iterations, with the approximation precision controlled by a test. Such an approach leads to substantial time savings in minimizing computationally expensive functions.

NUMERICAL EXPERIMENTS

In this section we show how our precision control algorithm can be used in conjunction with the Hooke-Jeeves optimization algorithm.

The objective is to fit four parameters of an air-to-water cooling coil computer simulation model such that the difference between simulated and measured coil air outlet temperature is minimal for a prescribed number of measurement points. The measurement data are the air and water inlet temperature, the air humidity ratio, the air and water mass flow, and the valve position of the throttle valve in the water circuit. There are 401 measurement data, equally spaced in time.

Simulation Model

The simulation model consists of a coupled system of nonlinear equations that is solved for 373 variables using Newton iterations. The model is static, simulated in SPARK 1.0.3 (LBNL and Ayres Sowell Associates Inc. 2003), and described in Xu and Haves (2001). For the range of measurement data, all model equations are once continuously differentiable, and the Jacobian matrix is non-singular. Therefore, it follows from the Implicit Function Theorem that the exact solution is a once continuously differentiable function.

The design parameters are the air and water side heat transfer coefficients and two parameters that define the valve characteristics.

We will control two precision parameters: the precision parameter for the Newton solver and the number of simulations that are used in the data fit.

Exact Cost Function

First, we define the exact cost function. Let $\tau \triangleq [0,\ 1]$ denote the normalized time interval over which the measurement took place. Let $T_m \colon [0,\ 1] \to \mathbb{R}$ be the linear interpolation of the measured coil air outlet temperature. For $x \in \mathbb{R}^n$ and $t \in \tau$, let $T_s(x,t) \in \mathbb{R}$ denote the exact solution of the system of equations that define the coil air outlet temperature, obtained by using linearly interpolated measurement data. For $e(x,t) \triangleq |T_m(t) - T_s(x,t)|$, we define the exact cost function

$$f(x) \triangleq \int_0^1 e(x,t)\, dt. \qquad (9)$$

Approximating Cost Functions

(9) cannot be evaluated because $T_s(x,t)$ can only be numerically approximated by an approximating solution $T_s^*(\varepsilon,x,t) \in \mathbb{R}$ with precision parameter $\varepsilon \in \mathbb{R}_+$, and the integral can only be approximated by a quadrature formula.

Let $\varepsilon^1 \in (0,\varepsilon_0^1] \subset \mathbb{R}_+$ denote the precision parameter of the Newton solver, and let $\varepsilon^2 \in (0,\varepsilon_0^2] \subset \mathbb{R}_+$ denote the time interval for the quadrature formula. In computing the approximating cost function, we have two levels of approximations: for $t \in \tau$, $e(x,t)$ is approximated by

$$e^*(\varepsilon,x,t) \triangleq |T_m(t) - T_s^*(\varepsilon^1,x,t)|, \qquad (10)$$

and for $0 < \varepsilon^2 \le \varepsilon_0^2 \le 1$ and $N(\varepsilon) \triangleq \lfloor 1/\varepsilon^2 \rfloor$, the integral (9) is approximated for $N(\varepsilon) > 1$ by

$$f^*(\varepsilon,x) \triangleq \qquad (11a)$$
$$\sum_{i=0}^{N(\varepsilon)-1} \frac{e^*(\varepsilon,x,i/N(\varepsilon)) + e^*(\varepsilon,x,(i+1)/N(\varepsilon))}{2N(\varepsilon)}.$$

If $N(\varepsilon) = 1$, we want to compute $e^*(\varepsilon,x,\cdot)$ only for $t = 0$, and set

$$f^*(\varepsilon,x) \triangleq e^*(\varepsilon,x,0). \qquad (11b)$$

It can be shown that there exist a $K \in (0,\infty)$ and an $\varepsilon_0 \in \mathbb{R}_+^2$ such that

$$|f^*(\varepsilon,x) - f(x)| \le K\, \|\varepsilon\|, \qquad (12)$$

for all $x \in \mathbb{R}^n$ and for all $\varepsilon \in \mathbb{R}_+^2$, with $\varepsilon \le \varepsilon_0$. Therefore, $\varphi(\cdot)$ in Assumption 1 is $\varphi(\varepsilon) = \|\varepsilon\|$.

Precision Control

To control $\varepsilon \in \mathbb{R}_+^2$, with $\varepsilon \le \varepsilon_0$, as a function of the mesh size factor $\Delta \in \mathbb{Q}_+$, we introduce $\rho \colon \mathbb{R}_+ \to \mathbb{R}_+^2$, with elements

$$\rho^1(\Delta) \triangleq \min\left(0.1,\ \varepsilon^1_{\min} \left(\frac{\Delta}{\Delta_{min}}\right)^{\alpha^1}\right), \qquad (13a)$$

$$\rho^2(\Delta) \triangleq \begin{cases} 1, & \text{if } \Delta = 1, \\ \varepsilon^2_{\min} \left(\frac{\Delta}{\Delta_{min}}\right)^{\alpha^2}, & \text{otherwise}, \end{cases} \qquad (13b)$$

where $\alpha^i > 1$, $\Delta_{min} \triangleq \min_{k\in\mathbb{N}}\{\Delta_k\}$ is the mesh size factor at which the optimization stops (Δ_{min} is known prior to the optimization), and $\varepsilon^i_{min} \in (0,\varepsilon^i_0)$ is the precision parameter for the last iterations. In (13a), we prevent $\rho^1(\Delta)$ from being too large since SPARK does not allow too large a value for $\varepsilon^1 = \rho^1(\Delta)$. In (13b), we set $\rho^2(1) = 1$ to use (11b) for the early iterations.

By (8), $\varphi(\rho(\Delta))/\Delta \to 0$, as $\Delta \to 0$.

Numerical Results

We will solve (1) using the Hooke-Jeeves optimization algorithm with adaptive precision function evaluations, where $f(\cdot)$ is approximated by (11), and ε is controlled using (13). In particular, we use the algorithm `GPSHookeJeeves` from GenOpt 2.0α (Wetter 2003). The optimization is done for different settings of α (see Table 1). For all adaptive precision optimizations, we use $\varepsilon_{min} = (1E{-}10, 1/400)^T$, $\Delta_0 = 1$ and 4 step reductions in which we set $\Delta_{k+1} = \Delta_k/2$. Hence, $\Delta_{min} = 1/16$, and we use all measurement data for the last iterations. For the fixed precision optimization, we set $\varepsilon = \varepsilon_{min}$ for all iterations.

As a measure of the computation time, we sum the number of function evaluations in SPARK for all iterations. This measure is proportional to the CPU time, but has the advantage that it is not influenced by other computation jobs that may run simultaneously. For our test suite, we normalize this measure against the number of function evaluations in SPARK for the case with fixed precision function evaluations. The computations are done on Linux RedHat 8.0 with an Intel Pentium III processor.

Table 1 shows the different settings for α, the normalized cost function value and the normalized measure for the CPU time. The minimum cost function value differs since the cost function has several local minima.

On average, our precision control scheme reduces the CPU time by 77%.

Figure 2 shows, for the optimization with adaptive precision function evaluations (with $\alpha = (9.14, 1.08)^T$) and fixed precision function evaluations, the cost function value as a function of the normalized measure of the CPU time. The horizontal axis in the right hand side of Figure 2 is logarithmic for better display of the early iterations. Below the axis we show when precision is increased (the different precisions are indicated by ε_k, $k \in \{0, \ldots, 4\}$). For this example, the precision control algorithm sets $\varepsilon_0 = (1.0E{-}01, 1)^T$, $\varepsilon_1 = (1.8E{-}02, 1/42)^T$, $\varepsilon_2 = (3.2E{-}05, 1/89)^T$, $\varepsilon_3 = (5.6E{-}08, 1/189)^T$, and $\varepsilon_4 = (1.0E{-}10, 1/400)^T$.

Table 1: Normalized cost function value at optimum and normalized measure for the CPU time for different precision control parameters α. All results are normalized against the optimization with fixed precision function evaluations.

(α^1, α^2)	normalized cost	normalized measure of CPU time
fixed precision	1.00	1.00
$(9.14, 1.08)$	0.92	0.18
$(9.14, 1.33)$	0.95	0.19
$(9.14, 2.16)$	0.97	0.27
$(9.14, 2.41)$	1.04	0.29
$(8.30, 1.08)$	0.92	0.18
$(8.30, 1.33)$	0.95	0.19
$(8.30, 2.16)$	0.97	0.27
$(8.30, 2.41)$	1.04	0.29
$(7.47, 1.08)$	0.92	0.18
$(7.47, 1.33)$	0.95	0.19
$(7.47, 2.16)$	0.97	0.27
$(7.47, 2.41)$	1.04	0.29
$(5.81, 1.08)$	0.92	0.18
$(5.81, 1.33)$	0.95	0.19
$(5.81, 2.16)$	0.97	0.28
$(5.81, 2.41)$	1.04	0.29
	average reduction	0.23

CONCLUSION

Thermal simulation programs construct discontinuous approximations to a usually continuously differentiable cost function. This can cause optimization methods to fail far from an optimal solution. In such cases, the economic potential that optimization offers is not utilized. To eliminate this problem, one needs to use high precision approximations that may require a prohibitively long computation time if used for all iterations.

We have presented a precision control scheme that uses low-cost, coarse precision approximations to the cost function when far from a solution, with the precision progressively increased as a solution is approached. For our scheme, convergence to a first order optimal point of the cost function can be proven even though the cost function is approximated by a family of discontinuous functions.

In the presented numerical experiments, our precision control scheme reduces the computation time by 77%.

ACKNOWLEDGMENTS

This research was supported by the Assistant Secretary for Energy Efficiency and Renewable Energy, Office of the Building Technologies Program of the U.S. Department of Energy, under Contract No. DE-AC03-76SF00098, and the National Science Foundation under Grant No. ECS-9900985.

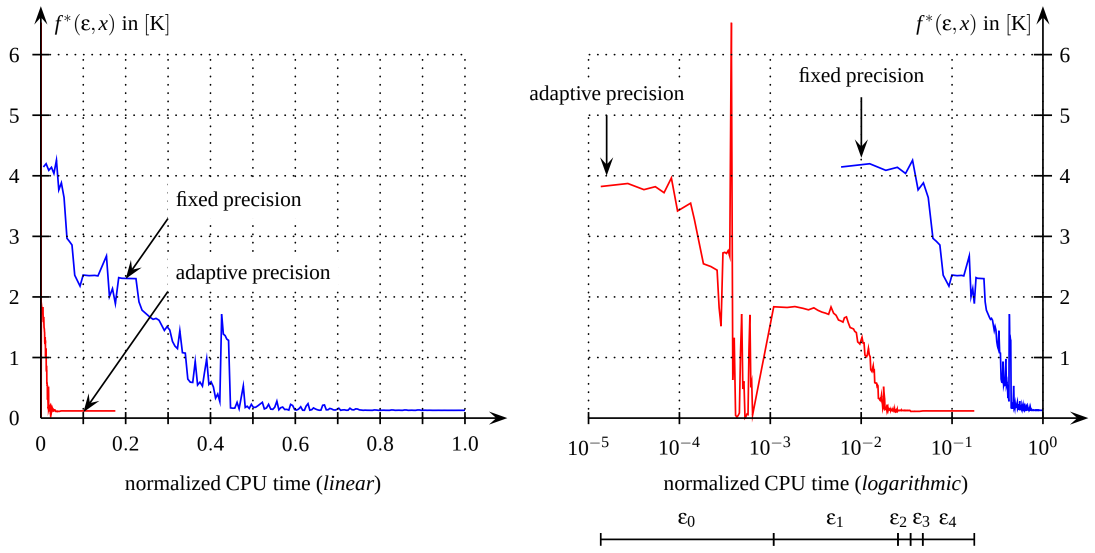

Figure 2: *Cost function value as a function of the normalized measure of the CPU time in linear scale (left-hand side) and logarithmic scale (right-hand side). Below the graph on the right we show for what intervals the precision ε has been kept constant.*

REFERENCES

Ascher, Uri M., and Linda R. Petzold. 1998. *Computer methods for ordinary differential equations and differential-algebraic equations*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Audet, Charles, and J. E. Dennis, Jr. 2003. “Analysis of Generalized Pattern Searches.” *SIAM Journal on Optimization* 13 (3): 889–903.

Bertsekas, Dimitri P. 1999. *Nonlinear Programming*. 2nd. Athena Scientific.

Brenan, K. E., S. L. Campbell, and L. R. Petzold. 1989. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. North-Holland.

Caldas, Luisa Gama, and Leslie K. Norford. 2002. “A design optimization tool based on a genetic algorithm.” *Automation in Construction* 11 (2): 173–184.

Coddington, Earl A., and Norman Levinson. 1955. *Theory of ordinary differential equations*. McGraw-Hill Book Company, Inc., New York-Toronto-London.

Crawley, Drury B., Linda K. Lawrie, Frederick C. Winkelmann, W.F. Buhl, Y. Joe Huang, Curtis O. Pedersen, Richard K. Strand, Richard J. Liesen, Daniel E. Fisher, Michael J. Witte, and Jason Glazer. 2001. “EnergyPlus: creating a new-generation building energy simulation program.” *Energy and Buildings* 33 (4): 443–457.

Hairer, E., and G. Wanner. 1996. *Solving ordinary differential equations. II.* 2nd. Springer series in computational mathematics. Berlin: Springer-Verlag.

Klein, S. A., J. A. Duffie, and W. A. Beckman. 1976. “TRNSYS – A Transient Simulation Program.” *ASHRAE Transactions*, vol. 82.

LBNL and Ayres Sowell Associates Inc. 2003. *SPARK, Reference Manual*. Berkeley, CA, USA: LBNL and Ayres Sowell Associates Inc.

Polak, Elijah. 1997. *Optimization, Algorithms and Consistent Approximations*. Volume 124 of *Applied Mathematical Sciences*. Springer Verlag.

Polak, Elijah, and Michael Wetter. 2003. “Generalized Pattern Search Algorithms with Adaptive Precision Function Evaluations.” Technical Report LBNL-52629, Lawrence Berkeley National Laboratory, Berkeley, CA.

Torczon, Virginia. 1997. “On the Convergence of Pattern Search Algorithms.” *SIAM Journal on Optimization* 7 (1): 1–25.

Wetter, Michael. 2001, August. “GenOpt – A Generic Optimization Program.” Edited by R. Lamberts, C. O. R. Negrão, and J. Hensen, *Proc. of the 7-th IBPSA Conference*, Volume I. Rio de Janeiro, Brazil, 601–608.

———. 2003. *GenOpt – Generic Optimization Program, User Manual*. 2.0α. Berkeley, CA, USA: Lawrence Berkeley National Laboratory.

Winkelmann, F.C., B. E. Birsdall, W. F. Buhl, K. L. Ellington, A. E. Erdem, J. J. Hirsch, and S. Gates. 1993, November. *DOE-2 Supplement, Version 2.1E*. Berkeley, CA, USA: Lawrence Berkeley National Laboratory.

Wright, Jonathan, and Heather Loosemore. 2001. "The Multi-Criterion Optimization of Building Thermal Design and Control." Edited by R. Lamberts, C. O. R. Negrão, and J. Hensen, *Proc. of the 7-th IBPSA Conference*, Volume I. 873–880.

Xu, Peng, and Philip Haves. 2001. "Library of component reference models for fault detection (AHU and chiller), Report to California Energy Commission." Technical Report, LBNL.

NOMENCLATURE

Conventions

1. Vectors are always column vectors, and their elements are denoted by superscripts.
2. Elements of a set or a sequence are denoted by subscripts.
3. $f(\cdot)$ denotes a function where $(\cdot)$ stands for the undesignated variables. $f(x)$ denotes the value of $f(\cdot)$ for the argument x. $f\colon A \to B$ indicates that the domain of $f(\cdot)$ is in the space A, and that the image of $f(\cdot)$ is in the space B.
4. We say that a function $f\colon \mathbb{R}^n \to \mathbb{R}$ is once continuously differentiable if $f(\cdot)$ is defined on $\mathbb{R}^n$, and if $f(\cdot)$ has a continuous derivative on $\mathbb{R}^n$.

Symbols

$f(\cdot)$	cost function
$f^*(\cdot,\cdot)$	approximating cost function
n	dimension of the independent parameter
q	dimension of the precision parameter of the numerical solvers
t	time
x	independent parameter
α	parameter to control how fast precision is increased
Δ_k	mesh size factor at k-th iteration
$a \in A$	a is an element of A
$A \subset B$	A is a subset of B
$\mathbb{N}$	$\{0,1,2,\ldots\}$
$\mathbb{Q}$	set of rational numbers
$\mathbb{Q}_+$	$\{q \in \mathbb{Q} \mid q > 0\}$
$\mathbb{R}$	set of real numbers
$\mathbb{R}^q_+$	$\{x \in \mathbb{R}^q \mid x^i > 0,\ i \in \{1,\ldots,q\}\}$
$\mathbb{Z}$	$\{\ldots,-2,-1,0,1,2,\ldots\}$
$\lfloor s \rfloor$	$\max\{k \in \mathbb{N} \mid k \le s\}$
$\triangleq$	equal by definition
e_i	unit vector along the i-th coordinate direction
$\|x\|$	L_2 norm of $x \in \mathbb{R}^n$, defined as $\|x\| \triangleq \big(\sum_{i=1}^n (x^i)^2\big)^{1/2}$